(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平11－134206

(43)公開日　平成11年(1999)５月21日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　9/46	３６０	Ｇ０６Ｆ　9/46	３６０Ｄ
13/00	３５５	13/00	３５５

審査請求　未請求　請求項の数11　ＯＬ　（全 11 頁）

(21)出願番号　特願平9－298064

(22)出願日　平成９年(1997)10月30日

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　平山　秀昭
東京都青梅市末広町２丁目９番地　株式会社東芝青梅工場内

(72)発明者　田中　邦典
東京都青梅市末広町２丁目９番地　株式会社東芝青梅工場内

(72)発明者　飯沼　哲也
東京都青梅市末広町２丁目９番地　株式会社東芝青梅工場内

(74)代理人　弁理士　大胡　典夫　（外１名）

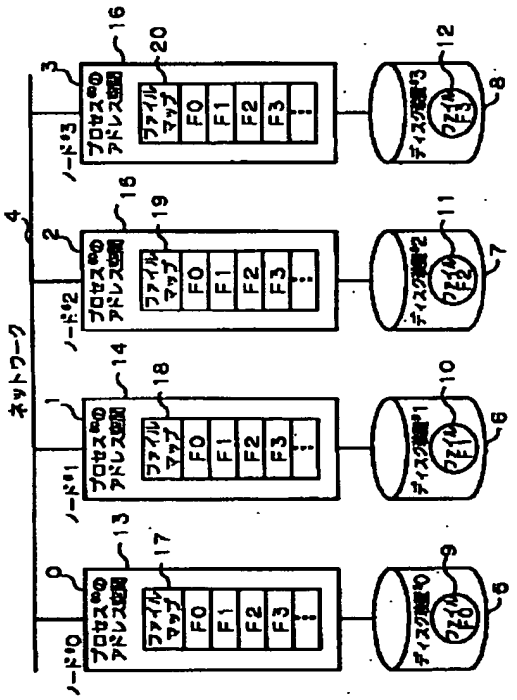
最終頁に続く

(54)【発明の名称】　分散共有リソース管理方法ならびに並列処理環境提供システム、及び同方法がプログラムされ記録される記録媒体

(57)【要約】

【課題】　本発明は、従来は高価な分散共有メモリ型マルチプロセッサ計算機やＭＰＰ計算機でしか実行できなかった、大規模のＤＷ／ＤＭの処理を、Ｗｉｎｄｏｗｓパソコン等の安価な計算機を多数接続した分散システム上で、容易にかつ高速に実行でき、かつ、分散共有リソースの概念をミドルウェアレベルで実現することを主な課題とする。

【解決手段】　本発明は、分散共有型のファィルマップ、セマフォ、プロセス生成、終了の各機能を持つ分散共有型のリソースを提供するシステムであって、他ノードｎ（１，２，３）からのセマフォ獲得要求が規定数以上蓄積されたら、ファイルマップ17（18，19，20）上で行なわれたデータ更新を他ノードに反映させ、データ更新の反映が終了したら保持し続けたセマフォを解放する一貫性保持プロトコルを持つ。



1

【特許請求の範囲】

【請求項1】　複数のノードから構成される分散システムで、任意ノード上のプロセスが任意ノード上のファイルを自身のアドレス空間にマップしてプロセッサのロード／ストア命令によってアクセスし、任意ノード上のプロセス間でリソースへのアクセスを排他制御するとともに、任意ノード上のプロセスが任意ノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させ、上記システム全体でプロセスがファイルマップ、セマフォを共有することにより協調処理を行う仮想的な並列処理環境を提供することを特徴とする分散共有リソース管理方法。

【請求項2】　複数のノードから構成される分散システムで、任意のノード上のプロセスが、任意ノード上のファイルを、自身のアドレス空間にマップしてプロセッサのロード／ストア命令によってアクセスし、任意のノード上のプロセスによるリソースへのアクセスを排他制御するとともに、任意ノード上のプロセスが、任意のノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させることのできる分散共有型のリソースを提供するシステムであって、異なるノード上のプロセスが同一のファイルを各々のプロセスのアドレス空間にマップして同時更新する場合に、（1）　プロセスが分散共有型のセマフォを獲得し、（2）　セマフォを獲得できたら分散共有型のファイルマップ上のデータを更新し、（3）　ファイルマップ上のデータの更新が済んでも獲得していたセマフォを、そのノードで保持し続け、（4）　他ノードからのそのセマフォの獲得要求を既定数以上受けたら、ファイルマップ上の更新データを他ノードに反映させ、（5）　更新データの反映が完了したら、保持し続けていたセマフォを解放することを特徴とする分散共有リソース管理方法。

【請求項3】　ファイルマップにアクセスするプロセスの数をノード毎にカウントし、異なるノード上のプロセスが、同一のファイルを各々のプロセスのアドレス空間にマップして同時更新する場合に、（1）　プロセスがセマフォを獲得し、（2）　セマフォを獲得できたら分散共有型のファイルマップ上のデータを更新し、（3）　ファイルマップの更新が済んでも獲得していたセマフォを、そのノードで保持し続け、（4）　他ノードからそのセマフォの獲得要求があった場合に、セマフォの獲得を要求したプロセスを待たせてしまうと、セマフォの獲得を要求したプロセスが実行中のノード上で実行可能なプロセスの数が、そのノード上に存在するプロセッサの数より少なくなってしまう場合には、ファイルマップ上の更新データを他ノードに反映させ、（5）　更新データの反映が完了したら、保持し続けていたセマフォを解放することを特徴とする分散共有リソース管理方法。

【請求項4】　異なるノード上のプロセスが、同一のファイルを各々のプロセスのアドレス空間にマップして、

2

同時更新する場合に、（1）　プロセスがセマフォを獲得し、（2）　セマフォを獲得できたら分散共有型のファイルマップ上のデータを更新し、（3）　ファイルマップ上のデータの更新が済んでも獲得していたセマフォを、そのノードで保持し続け、（4）　他ノードからそのセマフォの獲得要求があってから、既定時間以上経過した場合には、ファイルマップ上の更新データを他ノードに反映させ、（5）　更新データの反映が完了したら、保持し続けていたセマフォを解放することを特徴とする分散共有リソース管理方法。

【請求項5】　異なるノード上のプロセスが、同一のファイルを各々のプロセスのアドレス空間にマップして、同時更新する場合に、（1）　プロセスがセマフォを獲得し、（2）　セマフォを獲得できたら分散共有型のファイルマップ上のデータを更新し、（3）　ファイルマップ上のデータの更新が済んでも獲得していたセマフォを、そのノードで保持し続け、（4）　他ノードからそのセマフォの獲得要求があっても、セマフォの獲得を要求したプロセスを待たせておき、セマフォを保持しているノード上に実行可能なプロセスが無くなった場合に、ファイルマップ上の更新データを他ノードに反映させ、（5）　更新データの反映が完了したら、保持し続けていたセマフォを解放することを特徴とする分散共有リソース管理方法。

【請求項6】　複数のノードから構成される分散システムで、任意のノード上のプロセスが、任意ノード上のファイルを、自身のアドレス空間にマップしてプロセッサのロード／ストア命令によってアクセスし、任意のノード上のプロセスによるリソースへのアクセスを排他制御するとともに、任意ノード上のプロセスが、任意のノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させることのできる分散共有型のリソースを提供する並列処理環境システムであって、異なるノード上のプロセスが、同一のファイルを各々のプロセスのアドレス空間にマップして同時更新する場合に、プロセスが共有リソースにアクセスするのを排他制御するためのプロセスセマフォと、プロセスセマフォと1対1に対応し、プロセスセマフォを保持しているノードを示すノードセマフォを持つことを特徴とする分散共有リソースを用いた並列処理環境提供システム。

【請求項7】　プロセスが共有リソースをアクセスする場合に、それを保護するためのセマフォを一組のプロセスセマフォとノードセマフォで構成し、プロセスがプロセスセマフォの獲得要求を行った場合に、そのプロセスセマフォに対応するノードセマフォが、そのノードに保持されていて、かつ、そのノード上で他にそのプロセスセマフォを既に獲得しているプロセスがなければ、そのプロセスはそのプロセスセマフォを獲得して共有リソースへのアクセスを許可され、そのプロセスセマフォに対応したノードセマフォが、そのノードに保持されていな

ければ、他ノードにそのノードセマフォの獲得要求を出し、そのノードセマフォが獲得できたら、そのプロセスはそのプロセスセマフォを獲得することを特徴とする請求項6記載の分散共有リソースを用いた並列処理環境提供システム。

【請求項8】　プロセスがプロセスセマフォの解放を行った場合に、プロセスセマフォはそのプロセスから解放されるが、そのプロセスセマフォに対応したノードセマフォは、そのノード上で保持を継続することを特徴とする請求項6記載の分散共有リソースを用いた並列処理環境提供システム。

【請求項9】　複数のノードから構成される分散システムで、任意のノード上のプロセスが、任意ノード上のファイルを、自身のアドレス空間にマップしてプロセッサのロード/ストア命令によってアクセスし、任意のノード上のプロセスによるリソースへのアクセスを排他制御するとともに、任意ノード上のプロセスが、任意のノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させることのできる分散共有型のリソースを提供するシステムであって、
異なるノード上のプロセスが、同一のファイルを各々のプロセスのアドレス空間にマップして、並列更新する場合に、ファイルマップ上の更新データを他ノードに反映させて、そのノードで保持しているノードセマフォのうち、そのノードセマフォに対応しているプロセスセマフォがいかなるプロセスからも獲得されていないノードセマフォを全て解放する処理の前に、（1）　事前に更新データを含むページのリストを作成し、（2）　既定時間の間、それらのページが更新されたか否かを監視し、更新されたページは前記リストから削除し、（3）　既定時間経過後に前記リストに残ったページ上の更新データを、通常処理と並行して他ノードに反映させておくことにより、他ノードに更新データを反映する必要のあるページを減らしておき、（4）　前記ノードセマフォを解放する場合には、ファイルマップ上の残りの更新データを他ノードに反映させて、そのノードで保持しているノードセマフォのうち、そのノードセマフォに対応しているプロセスセマフォがいかなるプロセスからも獲得されていないものを全て解放することにより、ノードセマフォの解放時間を短縮することを特徴とする分散共有リソース管理方法。

【請求項10】　複数のノードから構成される分散システムで、任意のノード上のプロセスが、任意ノード上のファイルを、自身のアドレス空間にマップしてプロセッサのロード/ストア命令によってアクセスし、任意のノード上のプロセスによるリソースへのアクセスを排他制御するとともに、任意ノード上のプロセスが、任意のノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させることのできる分散共有型のリソースを提供し、プロセスが各リソースに対しアクセスするのを排他制御するためのプロセスセマフォと、プロセスセマフォと1対1に対応し、プロセスセマフォを保持しているノードを示すノードセマフォを持ち、プロセスセマフォの獲得を要求するステップと、獲得要求されたプロセスセマフォに対応するノードセマフォがそのノード上に保持されているか否か調べるステップと、保持されている場合には、そのプロセスセマフォを獲得し、保持していない場合には、他ノードに対してノードセマフォの獲得を要求するステップと、そのノードセマフォの獲得を待ち、ノードセマフォが獲得されたら、そのプロセスセマフォが他のプロセスに獲得されているか否かを調べるステップと、そのプロセスセマフォが他のプロセスに獲得されていれば、それが解放されるのを待ち、そのプロセスセマフォが他のプロセスから解放されたときにそのプロセスセマフォを獲得するステップとがプログラムされ記録されるコンピュータ読み取り可能な記録媒体。

【請求項11】　複数のノードから構成される分散システムで、任意のノード上のプロセスが、任意ノード上のファイルを、自身のアドレス空間にマップしてプロセッサのロード/ストア命令によってアクセスし、任意のノード上のプロセスによるリソースへのアクセスを排他制御するとともに、任意ノード上のプロセスが、任意のノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させることのできる分散共有型のリソースを提供し、プロセスが各リソースに対しアクセスするのを排他制御するプロセスセマフォと、プロセスセマフォと1対1に対応し、プロセスセマフォを保持しているノードを示すノードセマフォを持ち、データAを保護するプロセスセマフォAを獲得してデータAを更新するステップと、プロセスセマフォAを解放してデータBを保護するプロセスセマフォBを獲得してデータBを更新し、プロセスセマフォBを解放するステップと、プロセスセマフォA，Bに対応するノードセマフォA，Bを解放するために更新されたデータA，Bを他ノードに反映させ、ノードセマフォA，Bを解放するステップとがプログラムされ記録されるコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、分散システム上で実行する、データウェアハウス/データマイニングの処理に適用される、分散共有リソース管理方法ならびに並列処理環境提供システム、及び同方法がプログラムされ記録される記録媒体に関する。

【0002】

【従来の技術】近年の情報処理システムにおいて、SMP（Symmetric　Multi-Processor）、MPP（Massively　Parallel　Processor）は高性能化のための技術として定着してきた。単一のプロセッサ用に開発された

プログラムを並列化するのは自然な拡張であり、また、SMP、MPPの普及に伴いプログラマがそれ用の並列化作業に慣れ、それを受け入れやすい状況にある。

【0003】ところで、SMPは、バスのデータ転送能力のその拡張性に限界があり、一般的に数台から数十台のプロセッサ数が限界と言われている。その一方で分散メモリ型マルチプロセッサは、数百台を超すプロセッサの接続も可能である。しかし、分散メモリ型マルチプロセッサは、共有メモリを持たず、プログラマの視点からすると、単一プロセッサあるいはSMPとの間にギャップがあり、プログラム開発が容易ではない。

【0004】この単一プロセッサあるいはSMPと、分散メモリ型マルチプロセッサ間のギャップを埋めるものとして、分散共有メモリ型マルチプロセッサがある。分散共有メモリ型マルチプロセッサでは、プロセッサあるいはノード毎にメモリを持つが、アクセス時間の差はあるものの、自ノードのメモリのみならず、他ノードのメモリへのアクセスも可能にし、SMPの自然な拡張としてプロセッサの接続台数を大幅に増やしている。

【0005】現在のビジネス分野で、最も高い性能を必要とするアプリケーションとして、データウェアハウス（DW）とデータマイニング（DM）がある。これらのアプリケーションには、以下に列挙する（1）～（4）の特徴がある。

【0006】（1）小規模から超大規模まで幅広い要求がある。

【0007】（2）多プロセスを必要とする高い並列性を持つ。

【0008】（3）64ビットアドレス空間を期待する大容量データを扱う。

【0009】（4）データの参照は多いが、更新は少ない特に大規模なDW／DMを処理する場合には、上述した高価な分散共有メモリ型ノルチプロセッサ計算機やMPP計算機を利用していた。

【0010】

【発明が解決しようとする課題】大規模なDW／DMを処理する場合に利用される分散共有型マルチプロセッサ計算機やMPP計算機は非常に高価であり、Windows等のパソコンとの価格差は極めて大きい。即ち、大規模なDW／DMを処理するためには、多大な費用がかかるという問題があった。また、MPP計算機は、共有メモリを持たないため、DW／DMのプログラム開発が困難であるという問題があった。

【0011】本発明は上記事情に鑑みてなされたものであり、分散共有型のファィルマップ、セマフォ、プロセス生成／終了の各機能を持つ分散共有型のリソースを提供するシステムを実現することにより、従来は高価な分散共有メモリ型マルチプロセッサ計算機やMPP計算機でしか実行できなかった、大規模のDW／DMの処理を、Windowsパソコン等の安価な計算機を多数接続した分散システム上で、容易にかつ高速に実行することが可能な、分散共有リソースの概念をミドルウェアレベルで実現する、分散共有リソース管理方法ならびに並列処理環境提供システム、及び同方法がプログラムされ記録される記録媒体を提供することを目的とする。

【0012】

【課題を解決するための手段】本発明の分散共有リソース管理方法は、複数のノードから構成される分散システムで、任意のノード上のプロセスが、任意ノード上のファイルを、自身のアドレス空間にマップしてプロセッサのロード／ストア命令によってアクセスし、任意のノード上のプロセスによるリソースへのアクセスを排他制御するとともに、任意ノード上のプロセスが、任意のノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させることのできる分散共有型のリソースを提供し、異なるノード上のプロセスが同一のファイルを各々のプロセスのアドレス空間にマップして並列更新する場合に、プロセスが分散共有型のセマフォを獲得し、セマフォを獲得できたら分散共有型のファイルマップ上のデータを更新し、ファイルマップのデータの更新が済んでも獲得していたセマフォを、そのノードで保持し続け、他ノードからそのセマフォの獲得要求を既定数以上受けたら、ファイルマップ上の更新データを他ノードに反映させ、更新データの反映が完了したら、保持し続けていたセマフォを解放することを特徴とする。

【0013】また、異なるノード上のプロセスが、同一のファイルを各々のプロセスのアドレス空間にマップして、並列更新する場合に、ファイルマップ上の更新データを他ノードに反映させて、そのノードで保持しているノードセマフォのうち、そのノードセマフォに対応しているプロセスセマフォが獲得されていないものを全て解放する処理の前に、事前に更新データを含むページのリストを作成し、既定時間経過するまでの間、更に更新されたページは前記リストから削除し、既定時間経過後に前記リストに残ったページ上の更新データを、通常処理と並行して他ノードに反映させておくことによりセマフォ解放時に他ノードに反映する必要のある更新データを減らしておき、その後、ノードセマフォを解放する前にファイルマップ上の残りの更新データを他ノードに反映させて、そのノードで保持しているノードセマフォのうち、そのノードセマフォに対応しているプロセスセマフォが獲得されていないものを全て解放することも特徴とする。

【0014】本発明の並列処理環境提供システムは、複数のノードから構成される分散システムで、任意のノード上のプロセスが、任意ノード上のファイルを、自身のアドレス空間にマップしてプロセッサのロード／ストア命令によってアクセスし、任意のノード上のプロセスによるリソースへのアクセスを排他制御するとともに、任

意ノード上のプロセスが、任意のノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させることのできる分散共有型のリソースを提供し、異なるノード上のプロセスが、同一のファイルを各々のプロセスのアドレス空間にマップして並列更新する場合に、プロセスがリソースにアクセスするのを排他制御するプロセスセマフォと、プロセスセマフォと1対1に対応し、プロセスセマフォを保持しているノードを示すノードセマフォを持つことを特徴とする。

【0015】本発明の記録媒体は、コンピュータ読み取り可能であって、その記録媒体は、複数のノードから構成される分散システムで、任意のノード上のプロセスが、任意ノード上のファイルを、自身のアドレス空間にマップしてプロセッサのロード／ストア命令によってアクセスし、任意のノード上のプロセスによるリソースへのアクセスを排他制御するとともに、任意ノード上のプロセスが、任意のノード上に新たなプロセスを生成、もしくは任意ノード上で実行中のプロセスを終了させることのできる分散共有型のリソースを提供し、プロセスが各リソースに対しアクセスするのを排他制御するためのプロセスセマフォと、プロセスセマフォと1対1に対応し、プロセスセマフォを保持しているノードを示すノードセマフォを持ち、プロセスセマフォの獲得を要求するステップと、獲得要求されたプロセスセマフォに対応するノードセマフォがそのノード上に保持されているか否か調べるステップと、保持されている場合、そのプロセスセマフォを獲得し、保持していない場合、他ノードに対してノードセマフォの獲得を要求するステップと、そのノードセマフォの獲得を待ち、ノードセマフォが獲得されたら、そのプロセスセマフォが他のプロセスに獲得されているか否かを調べるステップと、そのプロセスセマフォが他のプロセスに獲得されていれば、それが解放されるのを待ち、そのプロセスセマフォが他の他のプロセスから解放されたときにそのプロセスセマフォを獲得するステップとがプログラムされ記録される。

【0016】また、データAを保護するプロセスセマフォAを獲得してデータAを更新するステップと、プロセスセマフォAを解放してデータBを保護するプロセスセマフォBを獲得してデータBを更新し、プロセスセマフォBを解放するステップと、プロセスセマフォA，Bに対応するノードセマフォA，Bを解放するために更新されたデータA，Bを他ノードに反映させ、ノードセマフォA，Bを解放するステップとがプログラムされ記録されることも特徴とする。

【0017】このことにより、従来は高価な分散共有メモリ型マルチプロセッサ計算機やMPP計算機でしか実行できなかった、大規模のDW／DMの処理を、例えば、Windows等の安価な計算機を多数接続した分散システム上で、容易にかつ高速に実行できるようになる。

【0018】

【発明の実施の形態】図1は本発明の実施形態を示すブロック図である。

【0019】図において、0、1、2、3は計算機のノードであり、各ノードにはディスク装置5、6、7、8が接続され、各ディスク装置5，6，7，8の中には9、10、11、12で示す、それぞれファイルF0（9），F1（10），F2（11），F3（12）が存在する。また、ノード0、1、2、3内には、13、14、15、16で示すプロセスのアドレス空間が存在し、各アドレス空間には17、18、19、20で示すファイルマップが存在し、各々がファイルF0，F1，F2，F3をマップしている。

【0020】図1に示す分散共有型のファイルマップ17，18，19では、複数のノード0，1，2，3上のプロセスが、複数のノード0，1，2，3のディスク装置5，6，7，8上に存在するファイル9，10，11，12を、自分のアドレス空間内にマップし、プロセッサのロード／ストア命令等でアクセスし、並列に更新する。

【0021】即ち、本発明で提供されるシステムは、分散システムの異なるノード上のプロセスが、任意のノードのディスク内に存在するファイルを、自身のアドレス空間にマップして並列に更新する機能を提供するものである。一見すれば、米国サンマイクロシステムズ社が提供するNFS（Network File System）が持つ機能を用いれば同じことが実現できるように思えるが実際にはそうではない。NFSを用いた場合、異なるノード上のプロセスが同一のファイルを各々のアドレス空間にマップして並列に更新した場合、データの一貫性が保持されない。これに対し本発明のシステムではデータの一貫性を保持できるていう点でNFSとは明らかに異なる。以下、その特徴を詳述する。

【0022】分散共有型のファイルマップ17，18，19，20では、ファイルマップ17，18，19，20上のデータを、複数のノード0，1，2，3上のプロセスが並列に更新するため、ファイルマップ17，18，19，20上のデータの一貫性を保持するためのプロトコルが必要となる。

【0023】本発明の実施形態では、以下のプロトコルに従うことにより、ファイルマップ17，18，19，20上のデータを、複数のノード0，1，2，3上のプロセスが並列に更新することを、可能にしている。

【0024】（1）プロセスセマフォを獲得する、（2）ファイルマップ上のデータを更新する、（3）プロセスセマフォを解放する、本発明の実施形態では、プロセスが各リソース（ファイルマップ上のデータ）に対しアクセスするのを排他制御するためのセマフォを一対のプロセスセマフォとノードセマフォによって実現する。プロセスセマフォは、プロセスがアクセスするリソ

ースを保証するためのセマフォで、ノードセマフォは、それと対をなすプロセスセマフォを保持しているノードを管理するためのセマフォである。

【００２５】図6は上述したプロセスセマフォとノードセマフォの関係を示す図であり、リソースAを保護するためのプロセスセマフォはＰＳ－Aで、ＰＳ－Aに対応するノードセマフォはＮＳ－Aである。同様に、リソースBを保護するためのプロセスセマフォはＰＳ－Bで、ＰＳ－Bに対応するノードセマフォはＮＳ－Bであり、リソースCを保護するためのプロセスセマフォはＰＳ－Cで、ＰＳ－Cに対応するノードセマフォはＮＳ－Cである。

【００２６】プロセスが、あるリソースにアクセスするためには、そのリソースを保護するプロセスセマフォの獲得を要求する。プロセスがプロセスセマフォの獲得要求を行った場合には、以下に列挙する手順に基づきプロセスセマフォの獲得が行われる。

【００２７】（１）そのプロセスセマフォに対応したノードセマフォが、そのノードに保持されていて、かつ、そのプロセスセマフォが他のプロセスに獲得されていなければ、そのプロセスはそのプロセスセマフォを獲得する。

【００２８】（２）そのプロセスセマフォに対応したノードセマフォが、そのノードに保持されていなければ、他ノードにそのノードセマフォの獲得要求を出し、そのノードセマフォが獲得できたら、そのプロセスはそのプロセスセマフォを獲得する。

【００２９】図2は上記（２）の場合の処理の様子をタイムテーブル上で示した図である。

【００３０】時刻ｔ１において、ノード０上のプロセス０が、リソースAにアクセスするために、リソースAを保護しているプロセスセマフォＰＳ－Aの獲得を要求している。この時、ノード０は、プロセスセマフォＰＳ－Aに対応するノードセマフォＮＳ－Aを保持していないので、他ノードにノードセマフォＮＳ－Aの獲得を要求する。

【００３１】時刻ｔ２において、ノードセマフォＮＳ－Aが他ノードから獲得され、それによりプロセス０は、プロセスセマフォＰＳ－Aを獲得できるようになる。プロセスセマフォＰＳ－Aを獲得したことにより、プロセス０はプロセスセマフォＰＳ－Aにより保護されているリソースAにアクセスできるようになる。

【００３２】時刻ｔ３において、プロセス０はリソースAのアクセスを終え、プロセスセマフォＰＳ－Aを解放するが、それに対応するノードセマフォＮＳ－Aは、そのままノード０上で保持し続けられる。

【００３３】図3は上記（１）の場合の処理の様子をタイムテーブル上で示した図である。

【００３４】時刻ｔ１において、ノード０上のプロセス０が、リソースAにアクセスするために、リソースAを保護しているプロセスセマフォＰＳ－Aの獲得を要求している。この時ノード０は、プロセスセマフォＰＳ－Aに対応するノードセマフォＮＳ－Aを保持している。しかもプロセスセマフォＰＳ－Aは、他のプロセスによって獲得されていない。従って、プロセス０は、すぐにプロセスセマフォＰＳ－Aを獲得する。

【００３５】時刻ｔ２において、プロセス０はリソースAのアクセスを終え、プロセスセマフォＰＳ－Aを解放するが、それに対応するノードセマフォＮＳ－Aは、ノード０上で保持し続けられる。

【００３６】図4はプロセスセマフォを獲得する際の処理の流れを示す図である。図中、４１はプロセスセマフォの獲得を要求するステップである。ステップ４２では獲得要求されたプロセスセマフォに対応するノードセマフォが、そのノード上に保持されているか否かを検査している。ここで、もしそのノードセマフォが保持されている場合には、ステップ４５でプロセスセマフォを即時に獲得する。しかしそのノードセマフォが保持されていない場合は、ステップ４３で他ノードにノードセマフォの獲得を要求し、ステップ４４でノードセマフォが獲得されるのを待つ。

【００３７】ノードセマフォが獲得されたら、ステップ４５で、そのプロセスセマフォが、他のプロセスに獲得されているか否かを調べる。もし、そのプロセスセマフォが、他のプロセスに獲得されているのであれば、それが解放されるのを待ち、そのプロセスセマフォが他のプロセスに獲得されていなくなったら、ステップ４６で、そのプロセスセマフォを獲得する。

【００３８】分散共有型のファイルマップでは、プロセスのアドレス空間にファイルをマップする。この際、更新データを含むページを監視するために、一旦全てのページを書き込み禁止状態でマップする。そして本来書き込み可能なページに対して書き込みが行われた場合、それに伴って発生するページフォールトの処理ルーチンで、そのページが更新されたことを記録し、更新前の状態でのページのコピーを保存し、そのページを書き込み可能状態でマップし直す。

【００３９】更新データを他ノードに反映する場合には、更新が記録されたページのみ、保存しておいた更新前の状態でのページのコピーと、現在の状態でのページを比較し、異なっている部分のみを、他ノードに反映させ、再びそのページを書き込み禁止状態にする。更新データを他ノードに反映させ、ノードセマフォを解放するための条件は、本発明の実施形態に従えば以下の何れかが成立した場合となっている。

【００４０】（１）他ノードからのノードセマフォ獲得要求が既定数以上溜まった。

【００４１】（２）ノードセマフォ獲得要求を行ったプロセスを待たせると、そのプロセス（ノードセマフォ獲得要求を行ったプロセス）を実行しているノード上の実

行可能プロセスの数が、そのノードに存在するプロセッサの数より少なくなる。

【0042】（3）他ノードからのノードセマフォ獲得要求があってから、既定時間を経過した。

【0043】（4）そのノード（ノードセマフォ獲得要求を出されたノード）で、実行可能なプロセスが無くなった。

【0044】図7にノードセマフォを解放する際の手順を示す。

【0045】PS－A，PS－Bは、データA，Bを保護するためのプロセスセマフォである。NS－A，NS－Bは、プロセスセマフォPS－A，PS－Bに対応したノードセマフォである。

【0046】図7ではデータA，Bを更新した後、それを保護するプロセスセマフォに対応するノードセマフォを解放するまでの手順を示している。ステップS71では、プロセスセマフォPS－Aを獲得している。ステップS72では、データAを更新している。ステップS73では、データAの更新を終え、プロセスセマフォPS－Aを解放しているが、ノードセマフォNS－Aは、そのノードで保持されたままになっている。ステップS74では、プロセスセマフォPS－Bを獲得している。ステップS75では、データBを更新している。ステップS76では、データBの更新を終え、プロセスセマフォPS－Bを解放しているが、ノードセマフォNS－Bは、そのノードで保持されたままになっている。

【0047】ステップS77では、プロセスセマフォPS－A，PS－Bに対応するノードセマフォNS－A，NS－Bを解放するために、更新されたデータA，Bを他ノードに反映させている。ステップS78では、更新データA，Bを他ノードに反映させたので、ノードセマフォNS－A，NS－Bを解放している。

【0048】図5（a）は、更新データを他ノードに反映させ、ノードセマフォを解放するまでの処理の様子をタイムテーブル上に示す図である。時刻t1までの間に、ページA，B，C，Dが更新されている。時刻t1では更新ページA，B，C，Dの更新データを他ノードに反映させ、それが完了した時刻t2において、現在このノードで保持されているノードセマフォのうち、そのノードセマフォに対応したプロセスセマフォが、どのプロセスからも獲得されていないものを全て解放する。

【0049】しかし図5（a）に示す手順では問題がある。本発明の実施形態に従えば、更新データを他ノードに反映させ、ノードセマフォを解放するタイミングをなるべく遅らせ、分散共有型のファイルマップ上のデータの一貫性保持のオーバヘッドを低減させている。しかしながらタイミングを遅らせている分だけ、一旦その状況になった場合には、速やかに更新データを他ノードに反映させ、ノードセマフォを解放したい。ところが逆に、タイミングを遅らせている分だけ、多くの更新データが溜り、更新データの他ノードへの反映に時間がかかってしまう。

【0050】そこで本発明では図5（b）に示す手順で更新データを他ノードに反映させている。図5（b）では、更新データを他ノードに反映させるタイミングは時刻t3である。しかしこの方法では、以前に更新された後、更新されなくなったページの更新データを、t3のタイミング以前に通常処理と並行して、他ノードに反映してしまうことにより、セマフォ解放時に、他ノードに反映する必要のあるページを減らしている。図5（b）では、ページA，Bは時刻t1には更新されたが、その後更新されなくなったページである。時刻t1においてそれまでに更新されたページのリストを作成し、その後時刻t2までの間それらのページが更に更新されないかどうかを、検査している。もし更に更新されたページがあれば、それらを前記のリストから外す。時刻t2に達した時点で、前記のリストに残っているページA，B上の更新データを、通常処理と並行して他ノードに反映させる。これにより時刻t3で更新データを他ノードに反映させるページは、C，Dの2つだけになっているため、更新データの他ノードへの反映時間が短くなり、時刻t4において、短時間で現在このノードで保持されているノードセマフォのうち、そのノードセマフォに対応したプロセスセマフォが、どのプロセスからも獲得されていないものを全て解放することができる。

【0051】尚、以上説明した本発明の実施形態は、単一のプロセッサあるいはSMPを各ノードとする分散システムにおいて実現されるものである。本発明は、このソフトウェアによって実現される分散共有リソースの概念をミドルウェアレベルで実現するものであり、具体的にこのソフトウェアは、フロッピーディスク、磁気ディスク、CD－ROM、MO、DVD等の記憶媒体に記録され、提供される。

【0052】

【発明の効果】以上説明のように本発明は、分散共有型のファイルマップ、セマフォ、プロセスの生成／終了の各機能を持つ分散共有型のリソースを提供するシステムであって、異なるノード上のプロセスが同一のファイルを各々のプロセスのアドレス空間にマップして並列更新する場合に、プロセスが分散共有型のセマフォを獲得し、セマフォを獲得できたら分散共有型のファイルマップ上のデータを更新し、ファイルマップ上のデータの更新が済んでも獲得していたセマフォを、そのノードで保持し続け、他ノードからそのセマフォの獲得要求を既定数以上受けたら、ファイルマップ上の更新データを他ノードに反映させ、更新データの反映が完了したら、保持し続けていたセマフォを解放する手順をミドルウェアレベルで実現するものであり、このことにより、従来は高価な分散共有メモリ型マルチプロセッサ計算機やMPP計算機でしか実行できなかった、大規模のDW／DMの

処理を、例えば、Ｗｉｎｄｏｗｓ等の安価な計算機を多数接続した分散システム上で、容易にかつ高速に実行できるようになる。

【図面の簡単な説明】

【図１】本発明の実施形態を示すブロック図、

【図２】本発明の実施形態の動作をタイムテーブル上に示した図、

【図３】本発明の実施形態の動作をタイムテーブル上に示した図、

【図４】本発明の実施形態の動作手順をフローチャートで示した図、

【図５】本発明の実施形態の動作をタイムテーブル上に示した図、

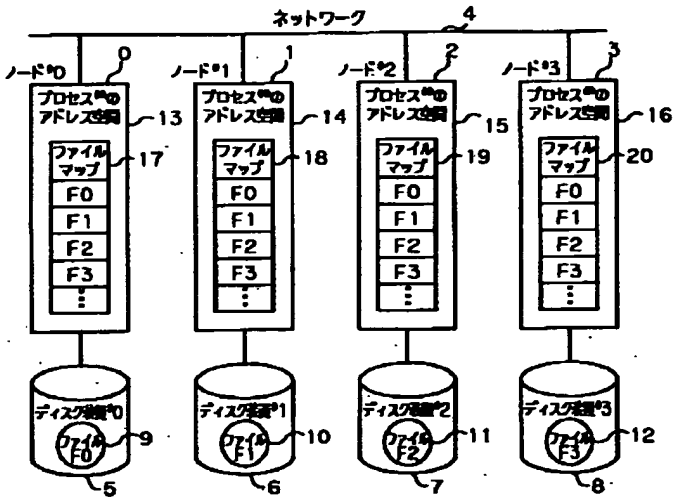
【図６】プロセスとセマフォの関係を示す図、

【図７】本発明の実施形態の動作手順をフローチャートで示した図、

【符号の説明】

０（１，２，３）…ノード、４…ネットワーク、５（６，７，８）…ディスク装置、９（１０，１１，１２）…ファイル、１３（１４，１５，１６）…プロセスｎのアドレス空間、１７（１８，１９，２０）…ファイルマップ。

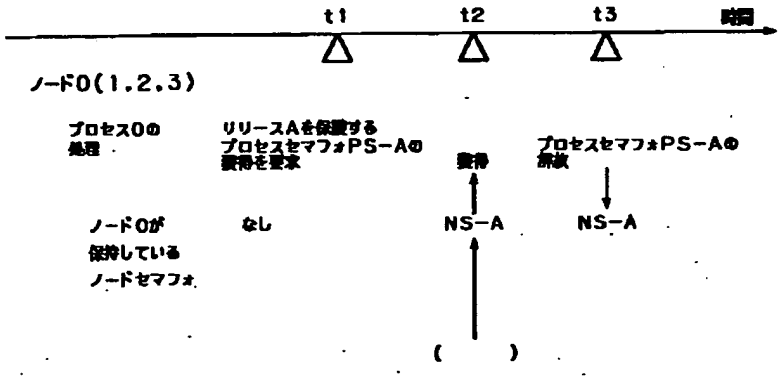
【図1】



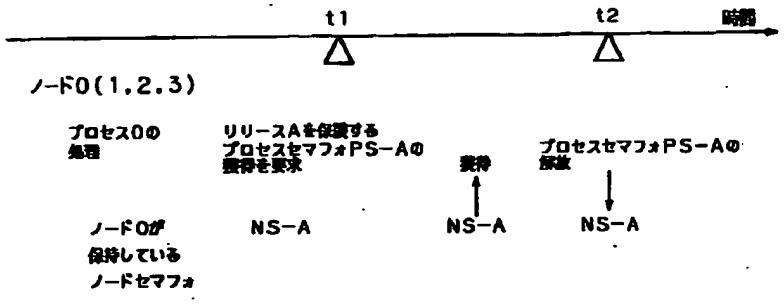
【図6】

プロセスセマフォ	ノードセマフォ	保護されるリソース
PS－A	NS－A	A
PS－B	NS－B	B
PS－C	NS－C	C
・	・	・
・	・	・
・	・	・
・	・	・

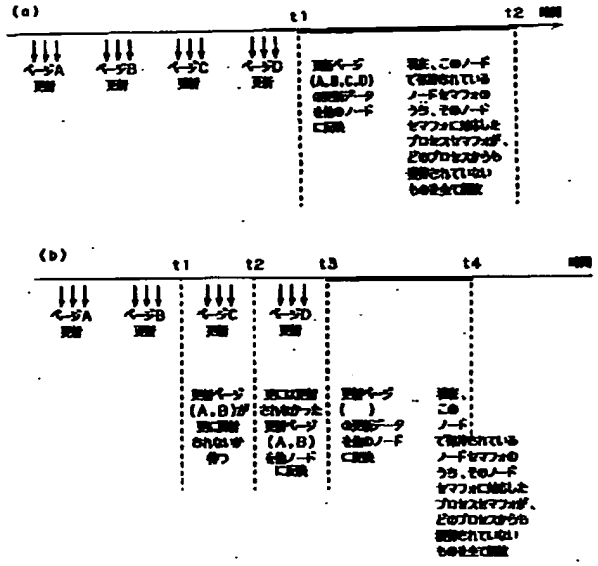
【図2】



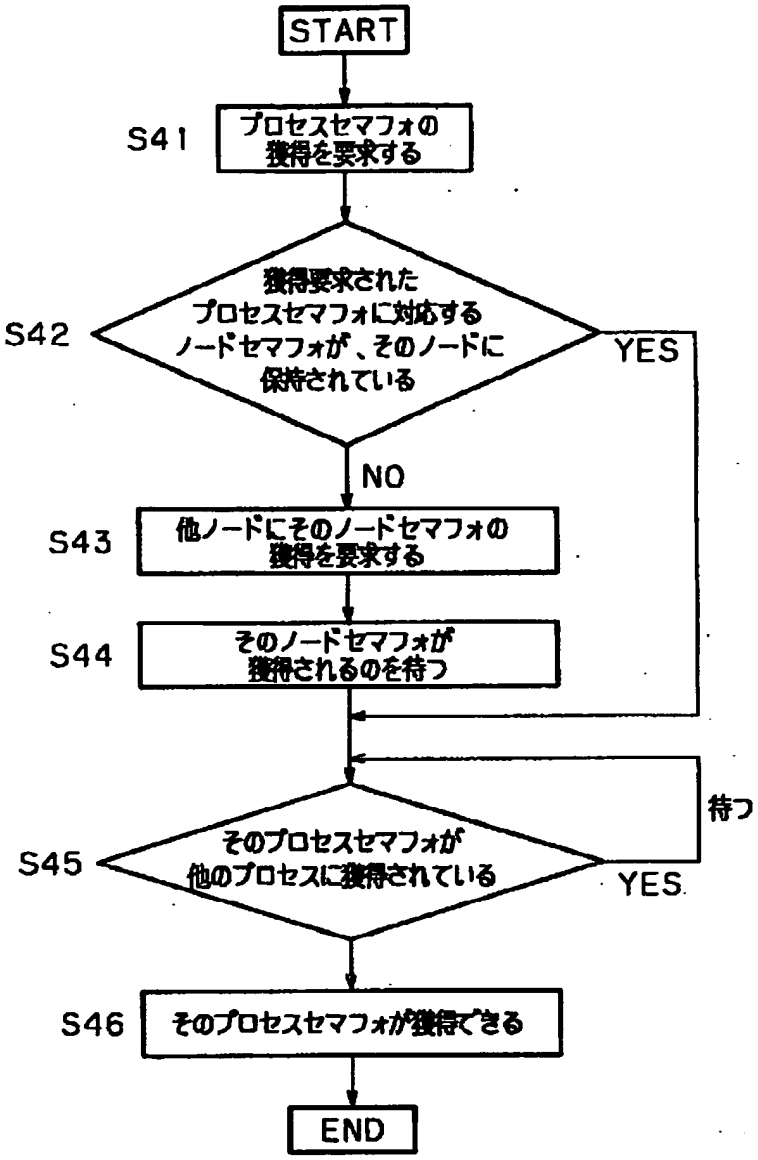
【図3】


t1
t2
時間
ノードO(1,2,3)
プロセスOの
処理
リリースAを保護する
プロセスセマフォPS－Aの
獲得を要求
獲得
プロセスセマフォPS－Aの
解放
ノードOが
保持している
ノードセマフォ
NS－A
NS－A
NS－A

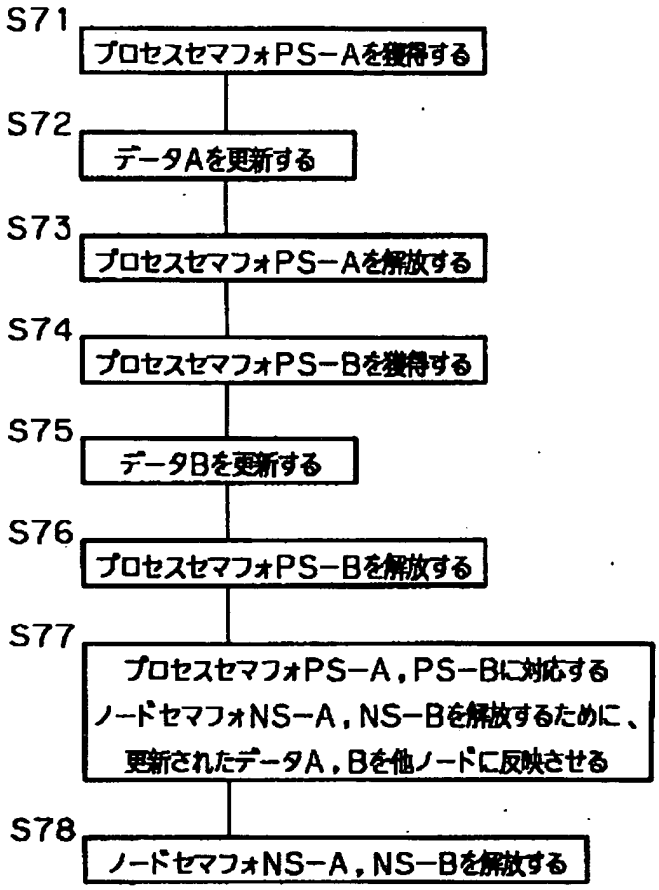
【図5】


(a)
t1
t2
時間
ページA
更新
ページB
更新
ページC
更新
ページD
更新
(b)
t1
t2
t3
t4
時間
ページA
更新
ページB
更新
ページC
更新
ページD
更新

【図4】


START
S41
プロセスセマフォの
獲得を要求する
S42
獲得要求された
プロセスセマフォに対応する
ノードセマフォが、そのノードに
保持されている
YES
NO
S43
他ノードにそのノードセマフォの
獲得を要求する
S44
そのノードセマフォが
獲得されるのを待つ
待つ
S45
そのプロセスセマフォが
他のプロセスに獲得されている
YES
S46
そのプロセスセマフォが獲得できる
END

【図7】



フロントページの続き

(72)発明者　白木原　敏雄
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

DERWENT-ACC-NO: 1999-362068

DERWENT-WEEK: 199931

TITLE: Distributed share resource management procedure for data warehousing or data mining in distributed information processing system - involves terminating current process and generating new process, when process shares file map and semaphore between several nodes

PATENT-ASSIGNEE: TOSHIBA KK[TOKE]

PRIORITY-DATA: 1997JP-0298064 (October 30, 1997)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 11134206 A	May 21, 1999	N/A	011	G06F 009/46

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO	APPL-DATE
JP 11134206A	N/A	1997JP-0298064	October 30, 1997

INT-CL (IPC): G06F009/46, G06F013/00

ABSTRACTED-PUB-NO: JP 11134206A

BASIC-ABSTRACT:

NOVELTY - When a process shares file map (17-20) and semaphore between several nodes, an imagination parallel- processing environment where a cooperation process is performed, is provided after terminating current process and generating new process on arbitrary node. DETAILED DESCRIPTION - The file (9-12) in arbitrary node (0-3) of distributed system is mapped to one's own address space (13-16), and is accessed by loading an instruction to processor. The exclusion control of access to resource is performed between the

processes on arbitrary node. INDEPENDENT CLAIMS are included for the following: parallel processing environment providing system; recording medium of distributed share resources management program

USE - For managing distributed share resources in data warehousing or data mining in distributed information processing system.

ADVANTAGE - Enables performing large scale process of data warehousing and data mining easily at high speed on distributed information processing system which connected many inexpensive computers of windows environment. DESCRIPTION OF DRAWING(S) - The figure shows block diagram of distributed information processing system. (0-3) Nodes; (9- 12) Files; (13-16) Address space of process; (17-20) File maps.

CHOSEN-DRAWING: Dwg.1/7

TITLE-TERMS: DISTRIBUTE SHARE RESOURCE MANAGEMENT PROCEDURE DATA WAREHOUSE DATA MINE DISTRIBUTE INFORMATION PROCESS SYSTEM TERMINATE CURRENT PROCESS GENERATE NEW PROCESS PROCESS SHARE FILE MAP SEMAPHORE NODE

DERWENT-CLASS: T01

EPI-CODES: T01-F02; T01-H;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N1999-270086